Int. Conf. on Computational Methods for Coupled Problems in Science and Engineering
COUPLED PROBLEMS 2009
B. Schrefler , E. Oñate and M. Papadrakakis (Eds)

DYNAMIC WORKFLOWS FOR MULTIPHYSICS DESIGN

TOAN NGUYEN*AND JEAN-ANTOINE DESIDERI†

* Institut National de Recherche en Informatique et Automatique (INRIA)
Centre de Recherche Grenoble Rhone-Alpes
655, av. de l'Europe
Montbonnot FR-38334 Saint Ismier, France
e-mail: Toan.Nguyen@inrialpes.fr, Web page: http://www-opale.inrialpes.fr

† INRIA Centre de Recherche Sophia-Antipolis Méditerrannée
06902 Sophia-Antipolis, France
e-mail: desideri@sophia.inria.fr - Web page: http://www-sop.inria.fr/opale

Key words: Multiphysics design, high-performance computing, parallel computing, workflows.

Summary. Large-scale multiphysics applications, e.g., aircraft flight simulation, require several layers for their effective implementation.The first layer includes the design of efficient methods for mathematical models, numeric problem solutions and data processing. This involves the optimization of complex application codes, that might include intricate and distributed execution tools. This includes the asynchronous execution of coordinated tasks executing in parallel on remotely connected environments, e.g., using grid middleware. The third layer includes sophisticated tools that allow the users to interact dynamically in explicit and coordinated ways to design new artefacts, e.g., workflow systems. This presentation is devoted to the third layer where sophisticated application codes are deployed and must be run cooperatively in heterogeneous, distributed and parallel environments. It is assumed here that the first and second layers are implemented and running to support the execution of the workflows. The paper focuses on some open challenges in deploying and running distributed workflows for mutiphysics design: resiliency, exception handling, dynamic user interactions and high-performance computing.

1 INTRODUCTION

The dynamic nature of complex collaborative design requires the implementation of sophisticated tools for the engineers to monitor and control the global process execution, to compare results, to rollback partial simulations and optimizations on-line and restart them using updated parameters[1,2]. This departs radically from past and current workflow engines which traditionnally run processes automatically. To achieve powerful interactions, and in addition to the usual abstract and concrete worflows scheme, the concept of shadow workflow is defined. It is used to implement efficient design protocols under full user control. Their implementation on distributed platforms and grid computing environments is required for existing workfow management systems. They must also provide effective rollback and restart procedures.

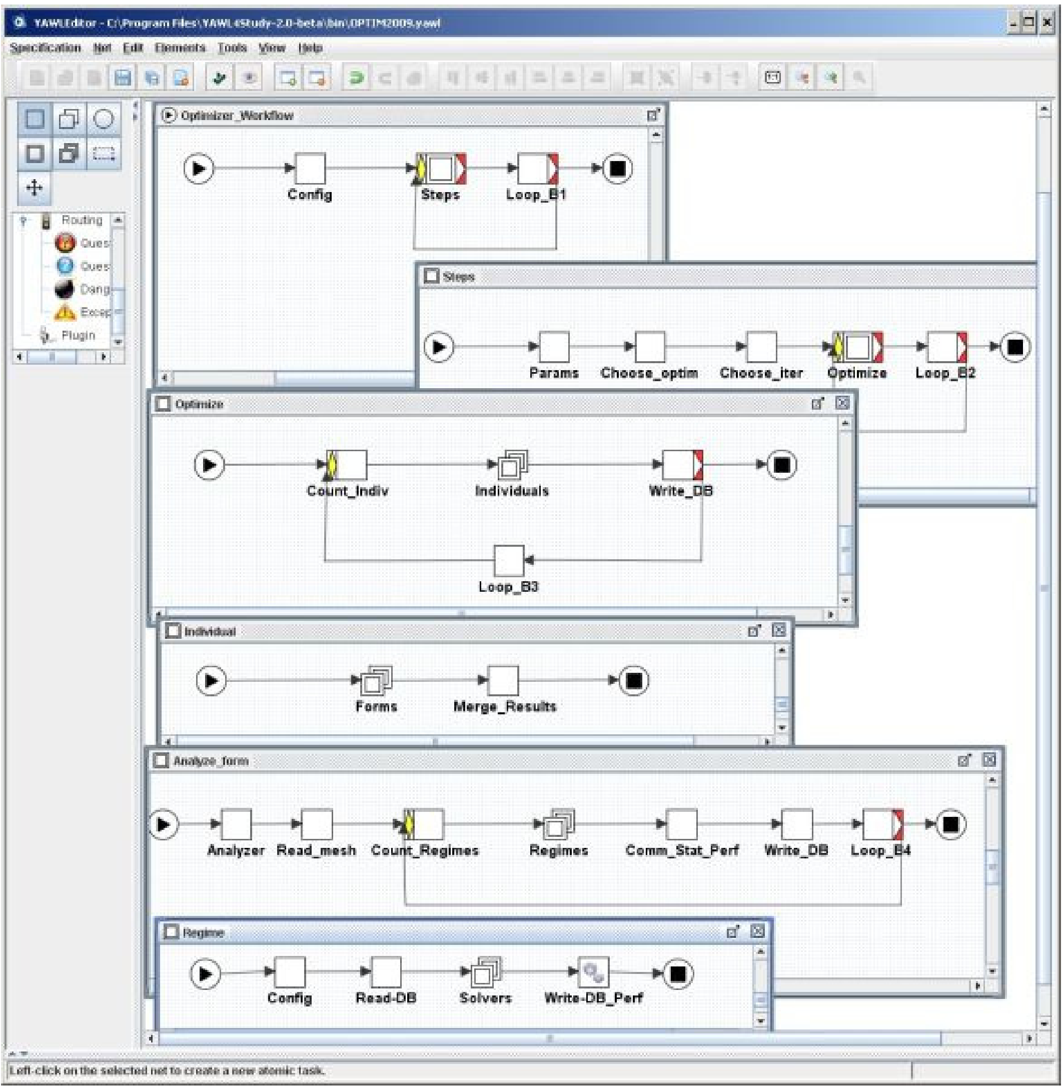

Figure 1. Wing optimization process definition using the YAWL workflow editor.

2 WORKFLOWS FOR SCIENTIFIC COMPUTING

The e-science communities are faced today with challenging opportunities concerning both the uptake of powerful computing technologies and the requirements for large-scale simulation and optimization applications[3,4]. This concerns innovative industry and academic sectors, including genomics, meteo and climate modeling, pharma, as well as engines efficiency and aerodynamics optimization for aircraft, trains and cars. Although used primarily for the automation of business processes, workflow systems have attracted much attention in these communities due to the complexity of multidiscipline applications, e.g., aeroacoustics, fluid-structure interaction for flutter prediction of aircraft wings, nuclear powerplant simulation, etc., and the simultaneaous need for seamless user interfaces.

A number of workflow systems and engines appeared to answer these needs[6], among which are: Taverna, Triana, Pegasus, Yawl, BPEL, BPEL4WS, MS WebSphere, Staffware, XPDL, etc. Some of them are distributed under open-source software licences. Although pursuing the same goal, e.g., automating processes in business and e-science domains, they usually provide specific variations on a common basis of control operators, i.e., task sequences, parallel tasks, task splits (OR and XOR splits) and joins (AND-joins), synchronization operators, with sometimes different semantics. Most notably, they usually support hierarchic task composition, multiple parallel instances of tasks, and sometimes interfaces with distributed and remote computing resources through Web Services. One of their obvious advantage is that they provide high-level and seamless user interfaces. They allow for a top-down definition of complex applications, formed by multiple and heterogeneous codes than can be synchronized and monitored.

Figure 1 is a screenshot of the YAWL workflow editor[5] which displays an example process definition for wing profile optimization. It shows the hierarchical composite structure of the workflow. It also shows the intricate parallelization of the process where the candidate solutions called here "Individuals" are evaluated in parallel for various wing profiles, called "Forms", at different flight "Regimes", which finally invoke various "Solvers" simultaneously in order to compare results.

3 REQUIREMENTS

Several requirements impact on workflow systems to support multidiscipline applications. Among them are: access to high-performance computing environments, resiliency, i.e., the ability to survive unpredictable hardware and software failures, exception-handling mechanisms in order to manage unexpected application behavior and fault-tolerance in order to tackle with reliability and availability issues, both hardware and software.

Another important aspect is user interactions. Indeed, optimization processes need engineers and designers to enter long-lasting trial and error loops as well as constraining parametrization steps. It is therefore a fundamental requirement that the automated processes be interrupted dynamically, i.e., during process execution, by the users, rolled-back and eventually restarted using updated parameters.

Multidiscipline applications are composed with various specialized codes that must interact according to specific data exchanges, synchronization plans and adaptation factors. They might require sophisticated management of multi-grid and time-step parameters, which are challenging goals for discipline experts. This is because interfacing different methods and tools pertaining to various disciplines makes multidscipline design, simulation and optimization a difficult endeavour.

Further, the scaling-up of the multidiscipline applications, e.g., 3D modeling and aeroacoustic analysis of full aircraft configurations, requires the access and use of high-performance interconnected computing resources, e.g., supercomputers, large PC-clusters, etc. The transparent access to heterogeneous computing environments is therefore a mandatory goal. It is made affordable nowadays using grid middleware and/or broadband networks.

The use of the high-performance environments and the collaborative nature of large multidiscipline projects makes also resiliency a stringent requirement. The ability of large-scale multidiscipline applications to survive to random hardware and software failures is required because the remote access to proprietary software and petascale data files is unavoidable in collaborative projects. Resiliency calls for sophisticated exception-handling mechanisms and fault-tolerance features, based on a variety of capabilities ranging from workflow shadowing to roll-back/ restart procedures. Shadowing is a feature which allows the logging of events and data sets that permit the tracability of processes and intermediate results, thus securing the provenance of data as well as process replays, restarts and resuming after parameter adjustements and crashes.

4 CONCLUSIONS

Workflow systems appeared long ago in the business arena to automate and effectively support administrative document processing. They recently attracted much attention in the e-science community because of their ability to support complex large-scale computations on distributed resources. Several workflow engines have been developed and are disseminated as open-source software, e.g., Pegasus, Taverna, Bonita …

The advent of distributed computing environments, e.g., grid middleware, have also emphasized their ability to support large-scale distributed simulation and optimization applications. These bear however specific properties that severely constrain the workflow systems, e.g., long duration runs, peta-scale data sets, dynamic parameterization interfaces. They require therefore specific functionalities such as resilience, sophisticated exception-handling mechanisms, fault-tolerance and user interactions that need particular attention.

Shadowing workflows that complement the usual abstract-concrete workflow implementations and event-based exception handling mechanisms are required to support these functionalities.

This work is supported by the AEROCHINA2 project of the EC, in the FP7 "Transport including Aeronautics" Program and by the OMD2 project ("Optimisation Multi-Discipline Distribuée") of the French National Research Agency ANR, program COSINUS.

REFERENCES

[1] T. Nguyên and J-A Désidéri. *Cooperative Design Workflows for Multiphysics Applications*, Proc. 5th Intl. CDVE Conf., Mallorca (Spain), LNCS 5220, Sept. 2008.
[2] T. Nguyên and J-A Désidéri. *Distributed Workflows for Multiphysics Applications*, Proc. 5th ECCOMAS Congress, Venice (Italy), July 2008.
[3] T. Nguyên, J-A Désidéri and J. Périaux. *Distributed Workflows for Multiphysics Applications in Aeronautics*, Proc. 12th Intl. Conf. on CSCW in Design (CSCWD 2008), Xi'an (China), April 2008.
[4] T. Nguyên and J. Periaux. *New collaborative working environments for multiphysics coupled problems*, Proc. ECCOMAS 2nd Intl. Conf. on Multiphysics Coupled Problems. Ibiza (Spain). May 2007.
[5] W. van der Aalst and al. *Design and implementation of the YAWL system*. Proc. 16th Intl. Conf. on Advanced Information Systems Engineering. CaiSE'2004. 2004.
[6] N. Russell, and al. *Workflow control flow patterns. A revised View*. Tech. Report. University of Technology Eindhoven (NL). 2006.